# Joint Inference and Disambiguation of Implicit Sentiments via Implicature Constraints

**Lingjia Deng**[1], **Janyce Wiebe**[1,2], **Yoonjung Choi**[2]
[1]Intelligent Systems Program, University of Pittsburgh
[2]Department of Computer Science, University of Pittsburgh
lid29@pitt.edu, wiebe@cs.pitt.edu, yjchoi@cs.pitt.edu

## Abstract

This paper addresses implicit opinions expressed via inference over explicit sentiments and events that positively/negatively affect entities (*goodFor/badFor*, *gfbf* events). We incorporate the inferences developed by implicature rules into an optimization framework, to jointly improve sentiment detection toward entities and disambiguate components of gfbf events. The framework simultaneously beats the baselines by more than 10 points in F-measure on sentiment detection and more than 7 points in accuracy on gfbf polarity disambiguation.

## 1 Introduction

Previous work in NLP on sentiment analysis has mainly focused on explicit sentiments. However, as noted in (Deng and Wiebe, 2014), many opinions are expressed implicitly, as shown by this example:

> Ex(1) The reform would lower health care costs, which would be a tremendous positive change across the entire health-care system.

There is an explicit positive sentiment toward the event of "reform lower costs". However, in expressing this sentiment, the writer also implies he is negative toward the "costs", since he's happy to see the costs being decreased. Moreover, the writer may be positive toward "reform" since it contributes to the "lower" event. Such inferences may be seen as opinion-oriented *implicatures* (i.e., defeasible inferences)[1].

We develop a set of rules for inferring and detecting implicit sentiments from explicit sentiments and events such as "lower" (Wiebe and Deng, 2014). In (Deng et al., 2013), we investigate such events, defining a ***badFor (bf)*** event to be an event that negatively affects the theme and a ***goodFor (gf)*** event to be an event that positively affects the theme of the event.[2] Here, "lower" is a bf event. According to their annotation scheme, ***goodFor/badFor (gfbf)*** events have NP agents and themes (though the agent may be implicit), and the polarity of a gf event may be changed to bf by a ***reverser*** (and vice versa).

The ultimate goal of this work is to utilize gfbf information to improve detection of the writer's sentiments toward entities mentioned in the text. However, this requires resolving several ambiguities: (Q1) Given a document, which spans are gfbf events? (Q2) Given a gfbf text span, what is its polarity, gf or bf? (Q3) Is the polarity of a gfbf event being reversed? (Q4) Which NP in the sentence is the agent and which is the theme? (Q5) What are the writer's sentiments toward the agent and theme, positive or negative? Fortunately, the implicature rules in (Deng and Wiebe, 2014) define dependencies among these ambiguities. As in Ex(1), the sentiments toward the agent and theme, the sentiment toward the gfbf event (positive or negative), and the polarity of the gfbf event (gf or bf) are all interdependent. Thus, rather than having to take a pipeline approach, we are able to develop an optimization framework which exploits these interdependencies to jointly resolve the ambiguities.

Specifically, we develop local detectors to analyze the four individual components of gfbf events, (Q2)-(Q5) above. Then, we propose an Integer Linear Programming (ILP) framework to conduct global

[1] Specifically, we focus on *generalized conversational implicature* (Grice, 1967; Grice, 1989).

[2] Compared to (Deng et al., 2013), we change the term "object" to "theme" as the later is more appropriate for this task.

*Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 79–88, Dublin, Ireland, August 23-29 2014.

inference, where the gfbf events and their components are variables and the interdependencies defined by the implicature rules are encoded as constraints over relevant variables in the framework. The reason we do not address (Q1) is that the gold standard we use for evaluation contains sentiment annotations only toward the agents and themes of gfbf events. We are only able to evaluate true hits of gfbf events. Thus, the input to the system is the set of the text spans marked as gfbf events in the corpus. The results show that, compared to the local detectors, the ILP framework improves sentiment detection by more than 10 points in F-measure and disambiguating gfbf polarity by more than 7 points in the accuracy, without any loss in accuracy for other two components.

## 2 Related Work

Most work in sentiment analysis focuses on classifying explicit sentiments and extracting explicit opinion expressions, holders and targets (Wiebe et al., 2005; Johansson and Moschitti, 2013; Yang and Cardie, 2013). There is some work investigating features that directly indicate implicit sentiments (Zhang and Liu, 2011; Feng et al., 2013). In contrast, we focus on how we can bridge between explicit and implicit sentiments via inference. To infer the implicit sentiments related to gfbf events, some work mines various syntactic patterns (Choi and Cardie, 2008), proposes linguistic templates (Zhang and Liu, 2011; Anand and Reschke, 2010; Reschke and Anand, 2011), or generates a lexicon of patient polarity verbs (Goyal et al., 2013). Different from their work, which do not cover all cases relevant to gfbf events, (Deng and Wiebe, 2014) defines a generalized set of implicature rules and proposes a graph-based model to achieve sentiment propagation between the agents and themes of gfbf events. However, that system requires all of the gfbf information (Q1)-(Q4) to be input from the manual annotations; the only ambiguity it resolves is sentiments toward entities. In contrast, the method in this paper tackles four ambiguities simultaneously. Further, as we will see below in Section 6, the improvement over the local detectors by the current method is greater than that by the previous method, even though it operates over the noisy output of local components automatically.

Different from pipeline architectures, where each step is computed independently, joint inference has often achieved better results. Roth and Yih (2004) formulate the task of information extraction using Integer Linear Programming (ILP). Since then, ILP has been widely used in various tasks in NLP, including semantic role labeling (Punyakanok et al., 2004; Punyakanok et al., 2008; Das et al., 2012), joint extraction of opinion entities and relations (Choi et al., 2006; Yang and Cardie, 2013), co-reference resolution (Denis and Baldridge, 2007), and summarization (Martins and Smith, 2009). The most similar ILP model to ours is (Somasundaran and Wiebe, 2009), which improves opinion polarity classification using discourse constraints in an ILP model. However, their work addresses discourse relations among explicit opinions in different sentences.

## 3 GoodFor/BadFor Event and Implicature

This work addresses sentiments toward, in general, states and events which positively or negatively affect entities. Deng et al. (2013) (hereafter DCW) identify a clear case that occurs frequently in opinion sentences, namely the gfbf events mentioned above. As defined in DCW, a *gf* event is an event that positively affects the theme of the event and a *bf* event is an event that negatively affects the theme. According to the annotation schema, *gfbf* events have NP agents and themes (though the agent may be implicit). In the sentence "President Obama passed the bill", the agent of the gf "passed" is "President Obama" and the theme is "the bill". In the sentence "The bill was denied", the agent of the bf "was denied" is implicit. The polarity of a gf event may be changed to bf by a *reverser* (and vice versa). For example, in "The reform will not worsen the economy," "not" is a reverser and it reverses the polarity from bf to gf.[3]

The constraints we encode in the ILP framework described below are based on implicature rules in (Deng and Wiebe, 2014). Table 1 gives two rule schemas, each of which defines four specific rules. In

[3] DCW also introduce *retainers*. We don't analyze retainers in this work since they do not affect the polarity of gfbfs, and only 2.5% of gfbfs have retainers in the corpus.

|  | s(gfbf) | gfbf | → | s(agent) | s(theme) |  | s(gfbf) | gfbf | → | s(agent) | s(theme) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | positive | gf | → | positive | positive | 3 | positive | bf | → | positive | negative |
| 2 | negative | gf | → | negative | negative | 4 | negative | bf | → | negative | positive |

Table 1: Rule Schema 1 & Rule Schema 3 (Deng and Wiebe, 2014)

the table, *s*($\alpha$) = $\beta$ means that the **writer's** sentiment toward $\alpha$ is $\beta$, where $\alpha$ is a gfbf event, or the agent or theme of a gfbf event, and $\beta$ is either *positive* or *negative*. P → Q means to infer Q from P.

Applying the rules to Ex(1): the writer expresses a positive sentiment ("positive") toward a bf event ("lower"), thus matching Case 3 in Table 1. We infer that the writer is positive toward the agent ("reform") and negative toward the theme ("costs"). Two other rule schemas (not shown) make the same inferences as Rule Schemas 1 and 3 but in the opposite direction. As we can see, if two entities participate in a gf event, the writer has the same sentiment toward the agent and theme, while if two entities participate in a bf event, the writer has opposite sentiments toward them. Later we use this observation in our experiments.

## 4 Global Optimization Framework

Optimization is performed over two sets of variables. The first set is *GFBF*, containing a variable for each gfbf event in the document. The other set is *Entity*, containing a variable for each agent or theme candidate. Each variable $k$ in *GFBF* has its corresponding agent and theme variables, $i$ and $j$, in *Entity*. The three form a triple unit, $\langle i, k, j \rangle$. The set *Triple* consists of each $\langle i, k, j \rangle$, recording the correspondence between variables in *GFBF* and *Entity*. The goal of the framework is to assign optimal labels to variables in *Entity* and *GFBF*. We first introduce how we recognize candidates for agents and themes, then introduce the optimization framework, and then define local scores that are input to the framework.

### 4.1 Local Agents and Theme Candidates Detector

We extract two agent candidates and two theme candidates for each gfbf event (one each will ultimately be chosen by the ILP model).[4] We use syntax, and the output of the SENNA (Collobert et al., 2011) semantic role labeling tool. SENNA labels the A0 (subject), A1 (object), and A2 (indirect object) spans for each predicate, if possible. To extract the *semantic agent* candidate: If SENNA labels a span as A0 of the gfbf event, we consider it as the semantic agent; if there is no A0 but A1 is labeled, we consider A1; if there is no A0 or A1 but A2 is labeled, we consider A2. To extract the *syntactic agent* candidate, we find the nearest noun in front of the gfbf span, and then extract any other word that depends on the noun according to the dependency parse. Similarly, to extract the *semantic theme* candidate, we consider A1, A2, A0 in order. To extract the *syntactic theme* candidate, the same procedure is conducted as for the syntactic agent, but the nearest noun should be after the gfbf. If there is no A0, A1 or A2, then there is only one agent candidate, *implicit* and only one theme candidate, *null*. We treat a *null* theme as an incorrect span in the later evaluations. If the two agent (theme) candidate spans are the same, there is only one candidate.

### 4.2 Integer Linear Programming Framework

We use Integer Linear Programming (ILP) to assign labels to variables. Variables in *Entity* will be assigned *positive* or *negative*, representing the writer's sentiments toward them. We may have two candidate agents for a gfbf and that we will choose between them. Thus, only one agent is assigned a *positive* or *negative* label; the other is considered to be an incorrect agent of the gfbf (similarly for the theme candidates). Each variable in *GFBF* will be assigned the label *gf* or *bf*. Optionally, it may also be assigned the label *reversed*. Label *gf* or *bf* is the polarity of the gfbf event; *reversed* is assigned if the polarity is reversed (e.g., for "not harmed", the labels are *bf* and *reversed*).

The objective function of the ILP is:

[4] This framework is able to handle any number of candidates. The methods we tried using more candidates did not perform as well - the gain in recall was offset by larger losses in precision.

$$\min_{u_{1gf},u_{1bf}\cdots}\left(-1*\sum_{i\in GFBF\cup Entity}\sum_{c\in L_i}p_{ic}u_{ic}\right)+\sum_{\langle i,k,j\rangle\in Triple}\xi_{ikj}+\sum_{\langle i,k,j\rangle\in Triple}\delta_{ikj} \tag{1}$$

subject to

$$u_{ic}\in\{0,1\},\forall i,c \qquad \xi_{ikj},\delta_{ikj}\in\{0,1\},\forall\langle i,k,j\rangle\in Triple \tag{2}$$

where $L_i$ is the set of labels given to $\forall i \in$ *GFBF* $\cup$ *Entity*. If $i \in$ *GFBF*, $L_i$ is {gf, bf, reversed} ({*gf, bf, r*}, for short). If $i \in$ *Entity*, $L_i$ is {positive, negative} ({*pos, neg*}, for short). $u_{ic}$ is a binary indicator representing whether the label $c$ is assigned to the variable $i$. When an indicator variable is 1, the corresponding label is selected. $p_{ic}$ is the score given by local detectors, introduced in the following sections. Variables $\xi_{ikj}$ and $\delta_{ikj}$ are binary slack variables that correspond to the gfbf implicature constraints of $\langle i,k,j\rangle$. When a given slack variable is 1, the corresponding triple violates the implicature constraints. Minimizing the objective function could achieve two goals at the same time. The first part ($-1*\sum_i\sum_c p_{ic}u_{ic}$) tries to select a set of labels that maximize the scores given by the local detectors. The second part ($\sum_{ikj}\xi_{ikj}+\sum_{ikj}\delta_{ikj}$) aims at minimizing the cases where gfbf implicature constraints are violated. Here we do not force each triple to obey the implicature constraints, but to minimize the violating cases. For each variable, we have defined constraints:

$$\sum_{c\in L_{GFBF'}}u_{kc}=1,\forall k\in GFBF \tag{3}$$

$$\sum_{\substack{i\in Entity\\ \langle i,k,j\rangle\in Triple}}\sum_{c\in L_{Entity}}u_{ic}=1,\forall k\in GFBF \tag{4}$$

$$\sum_{\substack{j\in Entity,\\ \langle i,k,j\rangle\in Triple}}\sum_{c\in L_{Entity}}u_{jc}=1,\forall k\in GFBF \tag{5}$$

where $L_{GFBF'}$ in Equation (3) is a subset of $L_{GFBF}$, consisting of {*gf, bf*}. Equation (3) means a gfbf must be either gf or bf. But it is free to choose whether it is being reversed. Recall that we have two agent candidates ($a1$,$a2$) for a gfbf. Thus we have four agent indicators in Equation (4): $u_{a1,pos}$, $u_{a1,neg}$, $u_{a2,pos}$ and $u_{a2,neg}$. Equation (4) ensures that three of them are 0 and one of them is 1. For instance, $u_{a1,pos}$ assigned 1 means that candidate $a1$ is selected to be the agent span and $pos$ is selected to be its polarity. In this way, the framework disambiguates the agent span and sentiment polarity simultaneously. (Similar comments apply for the theme candidates in Equation (5).)

According to the implicature rules in Table 1 in Section 3, the writer has the same sentiment toward entities in a gf relation. Thus, for each triple unit $\langle i,k,j\rangle$, the gf constraints are applied via the following:

$$|\sum_{i,\langle i,k,j\rangle}u_{i,pos}-\sum_{j,\langle i,k,j\rangle}u_{j,pos}|+|u_{k,gf}-u_{k,r}|<=1+\xi_{ikj},\forall k\in GFBF \tag{6}$$

$$|\sum_{i,\langle i,k,j\rangle}u_{i,neg}-\sum_{j,\langle i,k,j\rangle}u_{j,neg}|+|u_{k,gf}-u_{k,r}|<=1+\xi_{ikj},\forall k\in GFBF \tag{7}$$

We use $|u_{k,gf}-u_{k,r}|$ to represent whether this triple is gf. In Equation (6), if this value is 1, then the triple should follow the gf constraints. In that case, $\xi_{ikj}=0$ means that the triple doesn't violate the gf constraints, and $|\sum_i u_{i,pos}-\sum_j u_{j,pos}|$ must be 0. Further, in this case, $\sum_i u_{i,pos}$ and $\sum_j u_{j,pos}$ are constrained to be of the same value (both 1 or 0) – that is, entities $i$ and $j$ must be both positive or both not positive. However, if $\xi_{ikj}=1$, Equation (6) does not constrain the values of the variables at all. If $|u_{k,gf}-u_{k,r}|$ is 0, representing that the triple is not gf, then Equation (6) does not constrain the values of the variables. Similar comments apply to Equation (7).

In contrast, the writer has opposite sentiments toward entities in a bf relation.

$$|\sum_{i,\langle i,k,j\rangle}u_{i,pos}+\sum_{j,\langle i,k,j\rangle}u_{j,pos}-1|+|u_{k,bf}-u_{k,r}|<=1+\delta_{ikj},\forall k\in GFBF \tag{8}$$

$$|\sum_{i,\langle i,k,j\rangle}u_{i,neg}+\sum_{j,\langle i,k,j\rangle}u_{j,neg}-1|+|u_{k,bf}-u_{k,r}|<=1+\delta_{ikj},\forall k\in GFBF \tag{9}$$

We use $|u_{k,bf}-u_{k,r}|$ to represent whether this triple is bf. In Equation (8), if a triple is bf and the constraints are not violated, then $|\sum_i u_{i,pos}+\sum_j u_{j,pos}-1|$ must be 0. Further, in this case, $\sum_i u_{i,pos}$

|  | $u_{gf}$ | $u_{bf}$ | $u_r$ | $\lvert u_{gf} - u_r \rvert$ | $\lvert u_{bf} - u_r \rvert$ |  | $u_{gf}$ | $u_{bf}$ | $u_r$ | $\lvert u_{gf} - u_r \rvert$ | $\lvert u_{bf} - u_r \rvert$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 0 | 0 | 1 | 0 | C | 0 | 1 | 0 | 0 | 1 |
| B | 0 | 1 | 1 | 1 | 0 | D | 1 | 0 | 1 | 0 | 1 |

Table 2: Truth table of being reversed or not ($k$ is omitted)

and $\sum_j u_{j,pos}$ are constrained to be of the opposite value – that is, if entity $i$ is positive then entity $j$ must not be positive. Similar comments apply to Equation (9).

Note that above we use $|u_{k,gf} - u_{k,r}|$ and $|u_{k,bf} - u_{k,r}|$ to represent whether a triple is gf or bf. In Table 2, we show that they always take opposite values and that they are consistent with the actual polarities. In Table 2, Case A means the triple is gf and Case B means the triple is bf but it is reversed. In both cases, $|u_{gf} - u_r| = 1$, indicating that the triple should follow the gf constraints. Similarly for Case C and Case D to follow the bf constraints.

### 4.3 Local GoodFor/BadFor Score: $p_{k,gf}, p_{k,bf}$

We utilize a sense-level gfbf lexicon by (Choi et al., 2014). In total there are 6,622 gf senses and 3,290 bf senses. The gf lexicon covers 64% of the gf words in the corpus and the bf lexicon covers 42% of the bf words. We then look up the gfbf span $k$ in the gfbf lexicon. If $k$ only appears in the gf lexicon, then $p_{k,gf} = 1 - \epsilon$ and $p_{k,bf} = \epsilon$. Here $\epsilon = 0.0001$, to prevent there being any 0 scores in our computation. If $k$ only appears in the bf lexicon, then $p_{k,bf} = 1 - \epsilon$ and $p_{i,gf} = \epsilon$. If $k$ appears in both the gf and bf lexicon, and there are $a$ senses in the gf lexicon and $b$ senses in the bf lexicon, then $p_{k,gf} = a/(a + b)$ and $p_{k,bf} = b/(a + b)$. If $k$ is not in either lexicon, then $p_{k,gf} = p_{k,bf} = \epsilon$. If there is more than one word in the gfbf span, we take the maximum score.

### 4.4 Local Reversed Score: $p_{k,r}$

As introduced in Section 3, a reverser changes the polarity of a gfbf. First, we build reverser lexicons from Wilson's shifter lexicon (2008), namely the entries labeled as *genshifter*, *negation*, and *shiftneg*. We create two lexicons: one with the verbs and the other with the non-verb entries, excluding nouns, adjectives, and adverbs, since most non-verb reversers are prepositions or subordinating conjunctions. There are 219 reversers in the entire corpus; 134 (61.19%) are instances of words in one of the two lexicons. Based on the lexicon, we categorize reversers into three classes. Examples are shown below.

Ex(2) They will **not** be able to water down your coverage.
Ex(3) ... how a massive new bureaucracy will cut costs **without** hurting the old and the helpless.
Ex(4) The new law includes new rules to **prevent** insurance companies from overcharging patients.

**Negation:** An instance in this category is "not" in Ex(2). If any word in the gfbf span has a *neg* dependency relation according to the Stanford dependency parser, then we consider the gfbf to be negated (i.e., reversed). In this case the path between the negator and the gfbf is labeled *neg* and the length of the path is one.

**Other Non-Verb:** This category consists of words such as "without" in Ex(3) (others are "never" and "few", etc). These words lower the extent of the gfbf event. We look in the sentence for instances of words in the non-verb reverser lexicon, which are not tagged as noun, verb, adj, or adv. For any found, we examine the path in the dependency parse between the potential reverser and the gfbf span. If the path has at least one of *advmod*, *pcomp*, *cc*, *xcomp*, *nsubj*, *neg* and the length of the path is less than four (learnt from development set), the event is considered to be reversed.

**Verb:** In Ex(4), the verb "prevent" stops the gfbf event "overcharging" from happening. We call such words *Verb* reverser (others are "prohibit" and "ban", etc). We look in the sentence for instances of words in the verb reverser lexicon. For any that appear before the gfbf span in the sentence, if the path has at least one of *xcomp*, *pcomp*, *obj* and the length of the path is less than four, then the event is reversed. For the triple ⟨companies, overcharging, patients⟩ in Ex(4), though it is reversed by "prevent", the agent of the reverser, which is "law", is different from the agent of the gfbf, which is "companies", so the bf within the "overcharging" event is not reversed.[5] Though we extract the *Verb* reversers to evaluate the performance of recognizing a reverser, in the optimization framework, gfbf events with *Verb* reversers are not considered to be reversed, since almost all *Verb* reversers introduce new agents.

Different from other scores, $p_{k,r}$ could be negative. According to the heuristics above, the probability of a gfbf event being reversed decreases as the length of the path increases. We define $p_{k,r}$ so it is inversely proportional to the length of the path. Further, to make sense of a gfbf triple ⟨agent, gfbf, theme⟩, where, e.g., the local detectors label it ⟨pos, bf, pos⟩, the framework is choosing the smaller one from (a) $-1 * p_{k,r} * u_{k,r}$ (it has a reverser) versus (b) $1 * \xi_{ikj}$ (it is an exception to the rules). The framework assigns $u_{k,r} = 0$ and $\xi_{ikj} = 1$ if $-1 * p_{k,r} > 1$. It assigns $u_{k,r} = 1$ and $\xi_{ikj} = 0$ if $-1 * p_{k,r} <= 1$. For gfbf events which have *Negation* or *Other Non-verb* reversers, since we use the length four as a threshold in the heuristics above, we define $p_{k,r} = \frac{1}{d} - \frac{5}{4}$, so that $-1 * p_{k,r} = \frac{5}{4} - \frac{1}{d} > 1$ if $d > 4$. For gfbf events for which no reverser word appears in the sentence, or those which only have *Verb* reversers, $p_{k,r} = -1 * \frac{5}{4}$ (so $-1 * p_{k,r} > 1$), so that the framework chooses case (b) (choosing the gfbf event to be not reversed).

### 4.5 Local Sentiment Score: $p_{i,pos}, p_{i,neg}$

In the corpus of DCW, only the writer's sentiments toward the agents and the themes of gfbf events are annotated. Thus, since there are many false negatives of sentiments toward entities, the corpus does not support training a classifier. Therefore, we adopt the same local sentiment detector from (Deng and Wiebe, 2014), using available resources to detect writer's sentiments toward all agent and theme candidates.[6] The sentiment scores range from 0.5 to 1.

## 5 Co-reference In the Framework

So far the constraints in the framework are within a gfbf triple. Consider the following example:

Ex(5) **The reform** will decrease the healthcare costs and improve the medical qualify as expected.

The two gfbfs, "decrease" and "improve" have the same agent, "reform". Thus, if there is more than one gfbf in a sentence, and the path between the two gfbfs in dependency parse contains only *conj* or *xcomp*, and there is no other noun between the latter gfbf and the conjunction, we assume the two agents are the same and the sentiments toward them should be the same. Thus, for any $i, j \in$ *Entity*, if $i, j$ co-refer[7], or they are the same agent as described above, $Coref(i, j) = 1$ (otherwise 0). We add two more constraints, similar to the gf constraints in Equations (6) and (7), as shown in Equation (10) and (11). where $\nu_{ij}$ is a slack variable, $e(i)$ is the set of agent/theme candidates linked to the same gfbf as $i$ is. If $Coref(i, j) = 0$, Equations (10) and (11) do not constrain the variables. The objective function in Equation (12) is updated to incorporate these new constraints.

$$|\sum_{e(i)} u_{i,pos} - \sum_{e(j)} u_{j,pos}| + Coref(i,j) <= 1 + \nu_{ij}, \forall i,j \in Entity \tag{10}$$

$$|\sum_{e(i)} u_{i,neg} - \sum_{e(j)} u_{j,neg}| + Coref(i,j) <= 1 + \nu_{ij}, \forall i,j \in Entity \tag{11}$$

$$\min_{u_{1gf}, u_{1bf} \cdots} \left( -1 * \sum_{i \in GFBF \cup Entity} \sum_{c \in L_i} p_{ic} u_{ic} \right) + \sum_{\langle i,k,j \rangle \in Triple} \xi_{ikj} + \sum_{\langle i,k,j \rangle \in Triple} \delta_{ikj} + \sum_{i,j \in Entity} \nu_{ij} \tag{12}$$

## 6 Experiment and Performance

In this section we introduce the data we use, the baseline methods, the evaluations and the results. In addition, we give examples illustrating how opinion inference may improve performances.

[5] DCW defines here is a *triple chain*: ⟨law, prevent ⟨companies, overcharging, patients⟩⟩. The reverser is changing the polarity between "law" and "patients", but it does not change the polarity between "companies" and "patients".

[6] We use Opinion Extractor (Johansson and Moschitti, 2013) , opinionFinder (Wilson et al., 2005), MPQA subjectivity lexicon (Wilson et al., 2005), General Inquirer (Stone et al., 1966) and a connotation lexicon (Feng et al., 2013), to detect writer's sentiments toward all agent and theme candidates, and all gfbf events. We adopt Rule 1 and Rule 3 to infer from the sentiment toward event to the sentiment toward theme. Then we conduct a majority voting based on the results.

[7] We use the co-reference resolution system from (Stoyanov et al., 2010).

### 6.1 Experiment Data

We use the "Affordable Care Act" corpus of DCW, consisting of 134 online editorials and blogs. In total, there are 1,762 annotated triples, out of which 692 are gf or retainers and 1,070 are bf or reversers. From the writer's perspective, 1,495 noun phrases are annotated positive, 1,114 noun phrases are negative and the remaining 8 are neutral. This indicates that there are many opinions in the corpus. Out of 134 documents in the corpus, 3 do not have any annotation. 6 are used as a development set to develop the heuristics in Sections 4 and 5. We use the remaining 125 for the experiments.

### 6.2 Baseline Methods and Evaluation Metrics

We compare the output of the global optimization framework with the outputs of baseline systems built from the local detectors in Section 4. For the gfbf polarity and reverser ambiguities, the local detectors directly provide a disambiguation result. For the agent/theme span and sentiment ambiguities, the local sentiment detector assigns positive and negative scores to each candidate. The framework chooses among the combined options. Thus, for comparison, we build a baseline system that combines the outputs of the local agent/theme candidate detector and the local sentiment detector.

Recall from Section 4, a variable $k \in \mathit{GFBF}$ has two agent candidates, $a1$ and $a2 \in \mathit{Entity}$. Together there are four binary indicator variables: $u_{a1,pos}$, $u_{a1,neg}$, $u_{a2,pos}$ and $u_{a2,neg}$. Among these indicator variables whose corresponding local scores (e.g., $p_{a1,pos}$ is the score of $u_{a1,pos}$) are larger than 0.5, the baseline system (denoted *Local*) chooses the one with the largest local sentiment score. If there is a tie, it prefers the variable representing the semantic candidate. If there is still a tie, it chooses the variable representing the majority polarity (positive). If all the local scores of the four variables are 0.5 (neutral), *Local* fails to recognize any sentiment for that entity, so it assigns 0 to all the indicator variables. *Local+coref* takes the maximum local score of the entities if they co-ref, and assigns each entity the maximum score before disambiguation.

Another baseline, *Majority*, always chooses the semantic candidate and the majority polarity.

To evaluate the performance in detecting sentiment, we use precision, recall, and F-measure. We do not take into account any agent or theme manually annotated as neutral (there are only 8).

$$\text{P} = \frac{\text{\#(auto=gold \& gold!=neutral)}}{\text{\#auto!=neutral}} \qquad \text{Accuracy} = \text{R} = \frac{\text{\#(auto=gold \& gold!=neutral)}}{\text{\#gold!=neutral}} \qquad \text{F} = \frac{\text{2*P*R}}{\text{P+R}} \tag{13}$$

In the equations, *auto* is the system's output and *gold* is the gold-standard label from annotations. Since we don't take into account any *neutral* agent or theme, *#gold!=neutral* equals to all nodes in the experiment set. Thus accuracy is equal to recall. We only report recall here. Here we have two definitions of *auto=gold*: (1) **Strict** evaluation means that, by saying *auto=gold*, the agent/theme must have the same polarity and must be the same NP as the gold standard, and (2) **Relaxed** evaluation means the agent/theme has the same polarity as the gold standard, regardless whether the span is correct or not.

Note that according to DCW, an implicit agent isn't annotated with any sentiment. Thus, for an implicit agent in *gold*, if *auto* outputs the span "implicit", we treat it as a correct span with correct polarity, regardless what sentiment *auto* gives to it. If *auto* outputs any span other than "implicit", we treat it as a wrong span with wrong polarity, regardless of its sentiment as well. For the theme span, if *auto* outputs a "null" theme candidate, we treat it as a wrong span but we evaluate its sentiment according to *gold*.

To evaluate extracting candidate span, we use accuracy. The baseline for this task always chooses the semantic candidate. To evaluate gfbf polarity and reverser, we also use accuracy.

Note that although we evaluate the performance in different tasks separately, the framework resolves all the ambiguities at the same time.

### 6.3 Results

We report the performance results for **(A) sentiment detection** in Table 3, on two sets. One is the subset containing the agents and themes where *auto* has the correct spans with *gold*. The other is the set of all agents and themes. As shown in Table 3, *ILP* significantly improves performance, approximately 10-20 points on F-measure over different baselines. Though *Local* has a competitive precision with

| | | correct span subset | | | whole set, strict eval | | | whole set, relaxed eval | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F | P | R | F | P | R | F |
| 1 | ILP | 0.6421 | 0.6421 | 0.6421 | 0.4401 | 0.4401 | 0.4401 | 0.5939 | 0.5939 | 0.5939 |
| 2 | Local | 0.6409 | 0.3332 | 0.4384 | 0.4956 | 0.2891 | 0.3652 | 0.5983 | 0.3490 | 0.4408 |
| 3 | ILP+coref | 0.6945 | 0.6945 | 0.6945 | 0.4660 | 0.4660 | 0.4660 | 0.6471 | 0.6471 | 0.6471 |
| 4 | Local+coref | 0.6575 | 0.3631 | 0.4678 | 0.5025 | 0.3103 | 0.3836 | 0.6210 | 0.3834 | 0.4741 |
| 5 | Majority | 0.5792 | 0.5792 | 0.5792 | 0.3862 | 0.3862 | 0.3862 | 0.5462 | 0.5462 | 0.5462 |

Table 3: Performances of sentiment detection

*ILP*, it has a much lower recall. That means the local sentiment detector cannot recognize implicit sentiments toward most entities. But *ILP* is able to recognize more entities correctly. By adding *coref*, performance improves for both *ILP* and *Local*. In comparison to (Deng and Wiebe, 2014), our current method improves more in F-measure (2.43 points more) over local sentiment detector than the earlier work, even though the earlier work takes the manual annotations of all the gfbf information as input.

In terms of the other tasks: For **(B) agent/theme span**, the baseline achieves 66.67% in accuracy, compared to 68.54% and 67.10% for *ILP* and *ILP+coref*, respectively. For **(C) gfbf polarity**, the baseline has an accuracy of 70.68%, whereas *ILP* achieves 77.25% and *ILP+coref* achieves 77.47%, respectively, both 7 points higher. This improvement is interesting because it represents cases in which the optimization framework is able to ***infer*** the correct polarity even though the gfbf span is not recognized by the local detector (i.e., the span isn't in the gfbf lexicon). For **(D) reverser**, the baseline is 88.07% in accuracy. *ILP* and *ILP+coref* are competitive with the baseline: 89% and 88.07% respectively. Note that both our local detector and *ILP* surpass the majority class (not reversed) which has an accuracy of 86.60%.

Following (Akkaya et al., 2009), since ILP is unsupervised without multiple runs, we adopt McNemar's test to measure statistical significance of our improvements (Dietterich, 1998). In Table 3, the improvements in recalls of Line 1 over 2, Line 3 over 4, and Lines 1&3 over 5 are statistically significant at the $p < .001$ level. The improvements of Line 3 over 1 are statistically significant at the $p < .005$ level. For accuracy of gfbf polarity, the improvement is significant at the $p < .001$ level.

### 6.4 Examples

This sections gives simplified examples to illustrate how the framework can improve over the local detectors. The explicit sentiment clues referred to in this section are from MPQA lexicon.

Ex(6) The reform would curb skyrocketing costs in the long run.

The local sentiment detector assigns "costs" *negative* due to the single sentiment clue, "skyrocketing". Since the agent and theme are in a bf triple, and the writer is *negative* toward that theme, we can infer the writer is *positive* toward the agent. This illustrates how we improve recall on sentiments.

Ex(7) The supposedly costly reform will curb skyrocketing costs in the long run.

In Ex(7), agent "reform" is labeled *negative* because "costly" is a negative clue in the lexicon. ("supposedly" is not in it.) However, in Ex(7), it is actually positive. The agent's negative score is 0.6, and its positive score is 0.5 due to the absence of a positive clue. Since the theme is *negative* too, by the bf constraints, we **expect** to see a positive agent. If we were to assign *negative* to the agent, the objective function would have -0.6 subjectivity score and +1 in violation penalty, together giving +0.4. If we assign *positive*, the subjectivity score is -0.5, and there is no violation, resulting in a total score of -0.5. Thus, the framework correctly chooses the positive label. This shows how we can improve precision on sentiments.

Ex(8) The great reform will curb skyrocketing costs in the long run.

In this case, the agent is positive and the theme is negative. If the gfbf word "curb" is not in the lexicon, we could still infer its polarity. Given that the entities in the triple have different sentiments, to not violate the implicature rules, the framework will assign it *bf*, or assign it *gf* along with *reversed*. However, there is no reverser word in the sentence, so the reversed score $p_r = -\frac{5}{4}$. The framework will assign the reverser indicator $u_r = 0$, in order to avoid a gain in the objective function by $-1 * p_r * u_r$. Thus the framework assigns the label *bf* to “curb”. This is how the framework can improve the accuracy of recognizing gfbf polarity.

## 7 Conclusion

The ultimate goal of this work is to utilize gfbf information to improve detection of the writer’s sentiments toward entities mentioned in the text. Using an unsupervised optimization framework that incorporates gfbf implicature rules as constraints, our method improves over local sentiment recognition by almost 20 points in F-measure and over all sentiment baselines by over 10 points in F-measure. The global optimization framework jointly infers the polarity of gfbf events, whether or not they are reversed, which candidate NPs are the agent and theme, and the writer’s sentiments toward them. In addition to beating the baselines for sentiment detection, the framework significantly improves the accuracy of gfbf polarity disambiguation. This work not only automatically utilizes gfbf information to improve sentiment detection, it also proposes a framework for jointly solving various ambiguities related to gfbf events.

**Acknowledgement** This work was supported in part by DARPA-BAA-12-47 DEFT grant. We would like to thank the anonymous reviewers for their helpful feedback.

## References

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP ’09, pages 190–199, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pranav Anand and Kevin Reschke. 2010. Verb classes as evaluativity functor classes. In *Interdisciplinary Workshop on Verbs. The Identification and Representation of Verb Features*.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, Hawaii, October. Association for Computational Linguistics.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’06, pages 431–439, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoonjung Choi, Janyce Wiebe, and Lingjia Deng. 2014. Lexical acquisition for opinion inference: A sense-level lexicon of benefactive and malefactive events. In *5th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Dipanjan Das, André FT Martins, and Noah A Smith. 2012. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 209–217. Association for Computational Linguistics.

Lingjia Deng and Janyce Wiebe. 2014. Sentiment propagation via implicature constraints. In *Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2014)*.

Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. 2013. Benefactive/malefactive event and writer attitude annotation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 120–125, Sofia, Bulgaria, August. Association for Computational Linguistics.

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243, Rochester, New York, April. Association for Computational Linguistics.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

Song Feng, Jun Sak Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, Angust. Association for Computational Linguistics.

Amit Goyal, Ellen Riloff, and Hal Daum III. 2013. A computational model for plot units. *Computational Intelligence*, 29(3):466–488.

Herbert Paul Grice. 1967. Logic and conversation. The William James lectures.

Herbert Paul Grice. 1989. *Studies in the Way of Words*. Harvard University Press.

Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3).

André F. T. Martins and Noah a. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing - ILP '09*, pages 1–9, Morristown, NJ, USA. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1346. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Kevin Reschke and Pranav Anand. 2011. Extracting contextual evaluativity. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 370–374, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *CONLL*.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234, Suntec, Singapore, August. Association for Computational Linguistics.

P.J. Stone, D.C. Dunphy, M.S. Smith, and D.M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Coreference resolution with reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 156–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Janyce Wiebe and Lingjia Deng. 2014. An account of opinion implicatures. arXiv:1404.6491v1 [cs.CL].

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language ann. *Language Resources and Evaluation*, 39(2/3):164–210.

Theresa Wilson, Janyce Wiebe, , and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLP/EMNLP*, pages 347–354.

Theresa Wilson. 2008. *Fine-grained subjectivity analysis*. Ph.D. thesis, Doctoral Dissertation, University of Pittsburgh.

Bishan Yang and Claire Cardie. 2013. Joint Inference for Fine-grained Opinion Extraction. In *Proceedings of ACL*, pages 1640–1649.

Lei Zhang and Bing Liu. 2011. Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 575–580, Portland, Oregon, USA, June. Association for Computational Linguistics.